SQL PROJECT

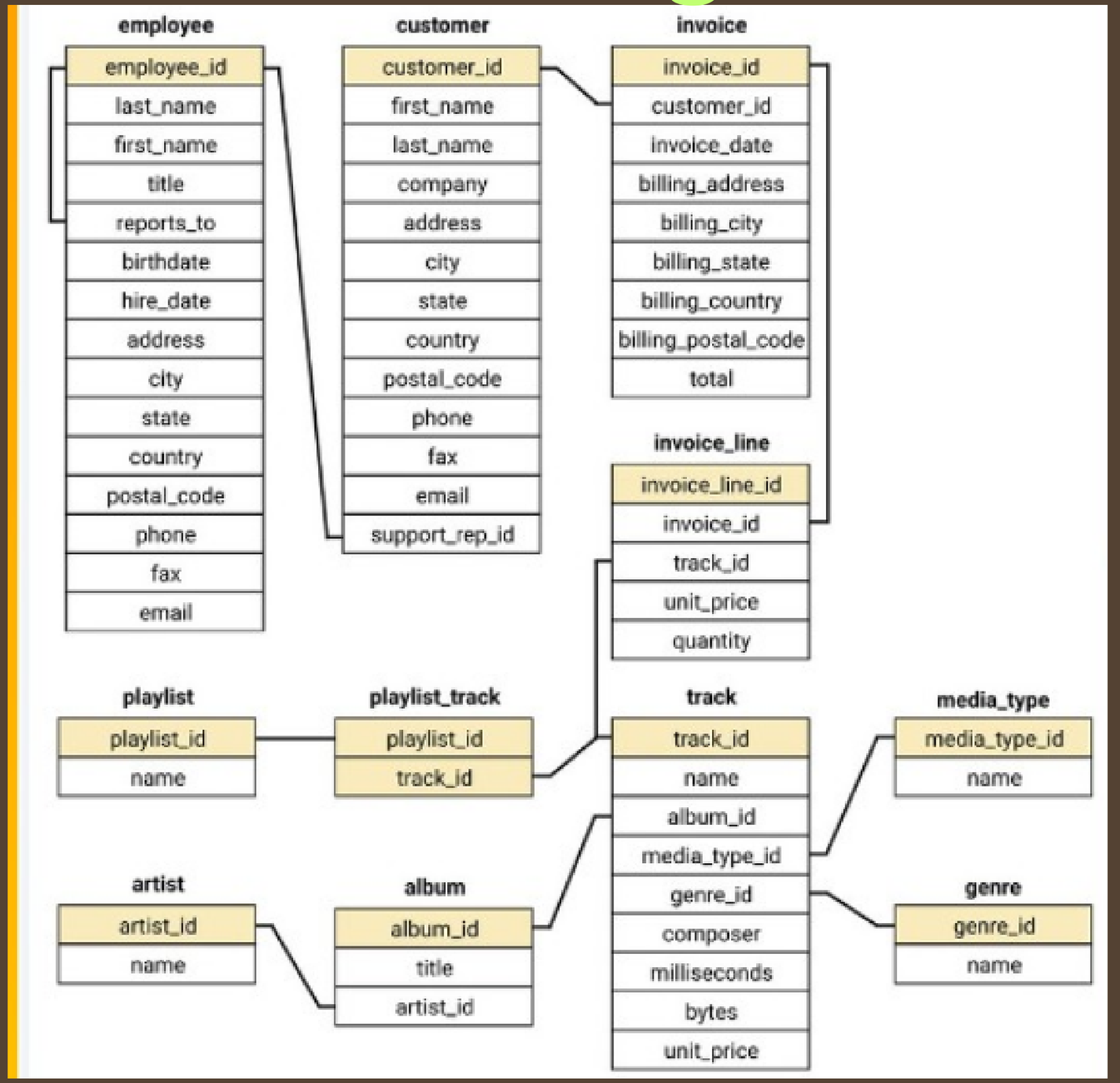
MUSIC STORE SALES ANALYSIS

Presented by : Mandeep Verma

PROJECT GOAL

The main objective of this project is to derive valuable insights from the dataset with SQL and help the store understand its business growth by answering simple questions

Schema Diagram



TABLE

- album
- artist
- customer
- employee
- genre
- invoice
- invoice_line
- media_type
- playlist
- playlist_track
- track

1. Who is the senior most employee based on job title?

```sql
SELECT title, last_name, first_name
FROM employee
ORDER BY levels DESC
LIMIT 1
```

Query | Query History | Messages | Notifications | Data Output

	title character varying (255)	last_name character varying (255)	first_name character varying (255)
1	Senior General Manager	Madan	Mohan

2. Which countries have the most Invoices?

Query | Query History | Messages | Notifications | Data Output

```
1
2  SELECT COUNT(*) AS c, billing_country
3  FROM invoice
4  GROUP BY billing_country
5  ORDER BY c DESC
```

Query | Query History | Messages | Notifications | Data Output

	c bigint	billing_country character varying (255)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany
6	30	Czech Republic
7	29	Portugal
8	28	United Kingdom
9	21	India
10	13	Chile
11	13	Ireland
12	11	Spain
13	11	Finland
14	10	Australia
15	10	Netherlands
16	10	Sweden
17	10	Poland
18	10	Hungary
19	10	Denmark
20	9	Austria
21	9	Norway
22	9	Italy

Total rows: 24 of 24 | Query complete 00:00:00.099

3. What are top 3 values of total invoice?

Query | Query History | Messages | Notifications | Data Output

```
1  SELECT total
2  FROM invoice
3  ORDER BY total DESC
4  limit 3
```

Query | Query History | Messages | Notifications | Data Output

	total numeric (10,2)
1	23.76
2	19.80
3	19.80

4. Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice totals

Query | Query History | Messages | Notifications | Data Output

```sql
SELECT SUM(total) AS Invoice_total,billing_city
FROM invoice
GROUP BY billing_city
ORDER BY Invoice_total DESC
limit 1;
```

Query | Query History | Messages | Notifications | Data Output

	invoice_total numeric	billing_city character varying (255)
1	273.24	Prague

5. Who is the best customer? The customer who has spent the most money will be declared the best customer. Write a query that returns the person who has spent the most money

Query | Query History | Messages | Notifications | Data Output

```sql
SELECT customer.customer_id, customer.first_name, customer.last_name, SUM(invoice.total) AS total
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id
ORDER BY total DESC
```

Query | Query History | Messages | Notifications | Data Output

	customer_id [PK] bigint	first_name character varying	last_name character varying	total numeric
1	5	František	Wichterlová	144.54

6. Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alphabetically by email starting with A

Query | Query History | Messages | Notifications | Data Output

```sql
SELECT DISTINCT email,first_name, last_name
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
WHERE track_id IN(
    SELECT track_id FROM track
    JOIN genre ON track.genre_id = genre.genre_id
    WHERE genre.name LIKE 'Rock'
)
ORDER BY email;
```

Query | Query History | Messages | Notifications | Data Output

	email character varying (100)	first_name character varying	last_name character varying
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis
11	eduardo@woodstock.com.br	Eduardo	Martins
12	ellie.sullivan@shaw.ca	Ellie	Sullivan
13	emma_jones@hotmail.com	Emma	Jones
14	enrique_munoz@yahoo.es	Enrique	Muñoz
15	fernadaramos4@uol.com.br	Fernanda	Ramos
16	fharris@google.com	Frank	Harris
17	fralston@gmail.com	Frank	Ralston
18	frantisekw@jetbrains.com	František	Wichterlová
19	ftremblay@gmail.com	François	Tremblay
20	fzimmermann@yahoo.de	Fynn	Zimmermann
21	hannah.schneider@yahoo.de	Hannah	Schneider
22	hholy@gmail.com	Helena	Holý

Total rows: 59 of 59 | Query complete 00:00:00.105

7. Let's invite the artists who have written the most rock music in our dataset. Write a query that returns the Artist name and total track count of the top 10 rock bands

Query | Query History | Messages | Notifications | Data Output

```sql
SELECT
    artist.artist_id,
    artist.name,
    COUNT(artist.artist_id) AS number_of_songs
FROM
    track
JOIN
    album ON album.album_id = track.album_id
JOIN
    artist ON artist.artist_id = album.artist_id
JOIN
    genre ON genre.genre_id = track.genre_id
WHERE
    genre.name LIKE 'Rock'
GROUP BY
    artist.artist_id,artist.name
ORDER BY
    number_of_songs DESC
LIMIT 10;
```

Query | Query History | Messages | Notifications | Data Output

	artist_id bigint	name character varying	number_of_songs bigint
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

8. Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first

Query | Query History | Messages | Notifications | Data Output

```sql
SELECT name,miliseconds
FROM track
WHERE miliseconds > (
    SELECT AVG(miliseconds) AS avg_track_length
    FROM track )
ORDER BY miliseconds DESC;
```

Query | Query History | Messages | Notifications | Data Output

	name character varying	miliseconds bigint
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677
10	Fire In Space	2926593
11	The Long Patrol	2925008
12	The Magnificent Warriors	2924716
13	The Living Legend, Pt. 1	2924507
14	The Gun On Ice Planet Zero, Pt. 2	2924341

9. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent

Query | Query History | Messages | Notifications | Data Output

```sql
WITH best_selling_artist AS (
    SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
    ROUND(SUM(invoice_line.unit_price*invoice_line.quantity))
    AS total_sales
    FROM invoice_line
    JOIN track ON track.track_id = invoice_line.track_id
    JOIN album ON album.album_id = track.album_id
    JOIN artist ON artist.artist_id = album.artist_id
    GROUP BY artist.artist_id, artist.name
    ORDER BY total_sales DESC
    LIMIT 1
)
SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name,
ROUND(SUM(il.unit_price*il.quantity)) AS amount_spent
FROM invoice i
JOIN customer c ON c.customer_id = i.customer_id
JOIN invoice_line il ON il.invoice_id = i.invoice_id
JOIN track t ON t.track_id = il.track_id
JOIN album alb ON alb.album_id = t.album_id
JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
GROUP BY c.customer_id, c.first_name, c.last_name, bsa.artist_name
ORDER BY 5 DESC;
```

Query | Query History | Messages | Notifications | Data Output

	customer_id bigint	first_name character varying	last_name character varying	artist_name character varying	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	28
2	38	Niklas	Schröder	Queen	19
3	3	François	Tremblay	Queen	18
4	34	João	Fernandes	Queen	17
5	53	Phil	Hughes	Queen	12
6	41	Marc	Dubois	Queen	12
7	47	Lucas	Mancini	Queen	11
8	33	Ellie	Sullivan	Queen	11
9	20	Dan	Miller	Queen	4
10	5	František	Wichterlová	Queen	4
11	23	John	Gordon	Queen	3
12	54	Steve	Murray	Queen	3
13	31	Martha	Silk	Queen	3
14	16	Frank	Harris	Queen	2
15	17	Jack	Smith	Queen	2
16	24	Frank	Ralston	Queen	2
17	30	Edward	Francis	Queen	2
18	35	Madalena	Sampaio	Queen	2
19	36	Hannah	Schneider	Queen	2
20	11	Alexandre	Rocha	Queen	2
21	8	Daan	Peeters	Queen	2
22	42	Wyatt	Girard	Queen	2
23	44	Terhi	Hämäläinen	Queen	2
24	1	Luís	Gonçalves	Queen	2
25	48	Johannes	Van der Berg	Queen	2
26	49	Stanisław	Wójcik	Queen	2
27	52	Emma	Jones	Queen	2
28	57	Luis	Rojas	Queen	2

Total rows: 43 of 43 | Query complete 00:00:00.086

10. We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where the maximum number of purchases is shared return all Genres

ry Query History Messages Notifications Data Output

```sql
WITH popular_genre AS
(
    SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
    ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
    FROM invoice_line
    JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
    JOIN customer ON customer.customer_id = invoice.customer_id
    JOIN track ON track.track_id = invoice_line.track_id
    JOIN genre ON genre.genre_id = track.genre_id
    GROUP BY customer.country, genre.name, genre.genre_id
    ORDER BY customer.country ASC, 1 DESC
)
SELECT * FROM popular_genre WHERE RowNo <= 1
```

Query Query History Messages Notifications Data Output

	purchases bigint	country character	name character varying	genre_id bigint	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1
11	211	France	Rock	1	1
12	194	Germany	Rock	1	1
13	44	Hungary	Rock	1	1
14	102	India	Rock	1	1
15	72	Ireland	Rock	1	1
16	35	Italy	Rock	1	1
17	33	Netherlands	Rock	1	1
18	40	Norway	Rock	1	1
19	40	Poland	Rock	1	1
20	108	Portugal	Rock	1	1
21	46	Spain	Rock	1	1
22	60	Sweden	Rock	1	1
23	166	United Kingdom	Rock	1	1
24	561	USA	Rock	1	1

11.Write a query that determines the customer that has spent the most on music for each country. Write a query that returns the country along with the top customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount

y Query History Messages Notifications Data Output

```sql
WITH Customter_with_country AS (
        SELECT customer.customer_id,first_name,last_name,billing_country,
        SUM(total) AS total_spending,
        ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
        FROM invoice
        JOIN customer ON customer.customer_id = invoice.customer_id
        GROUP BY customer.customer_id,first_name,last_name,billing_country
        ORDER BY billing_country  ASC,5 DESC)
SELECT * FROM Customter_with_country WHERE RowNo <= 1
```

Query Query History Messages Notifications Data Output

	customer_id bigint	first_name character varying	last_name character varying	billing_country character varying (255)	total_spending numeric	rowno bigint
1	56	Diego	Gutiérrez	Argentina	39.60	1
2	55	Mark	Taylor	Australia	81.18	1
3	7	Astrid	Gruber	Austria	69.30	1
4	8	Daan	Peeters	Belgium	60.39	1
5	1	Luís	Gonçalves	Brazil	108.90	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.02	1
8	5	František	Wichterlová	Czech Republic	144.54	1
9	9	Kara	Nielsen	Denmark	37.62	1
10	44	Terhi	Hämäläinen	Finland	79.20	1
11	42	Wyatt	Girard	France	99.99	1
12	37	Fynn	Zimmermann	Germany	94.05	1
13	45	Ladislav	Kovács	Hungary	78.21	1
14	58	Manoj	Pareek	India	111.87	1
15	46	Hugh	O'Reilly	Ireland	114.84	1
16	47	Lucas	Mancini	Italy	50.49	1
17	48	Johannes	Van der Berg	Netherlands	65.34	1
18	4	Bjørn	Hansen	Norway	72.27	1
19	49	Stanisław	Wójcik	Poland	76.23	1
20	34	João	Fernandes	Portugal	102.96	1
21	50	Enrique	Muñoz	Spain	98.01	1
22	51	Joakim	Johansson	Sweden	75.24	1

Total rows: 24 of 24 Query complete 00:00:00.117

Thank’s For Watching